REMARKS

Reconsideration and allowance of the present application are respectfully requested. Claims 1-2 and 4-42 are currently pending in this application.

Regarding Typographical Corrections in the Specification

Revisions were made to pages 15 and 26 of the specification to correct typographical errors discovered upon review of the application.

Regarding the 35 U.S.C. § 101 Rejection

The Office Action rejects claims 1-7 under 35 U.S.C. § 101 because the claimed invention is alleged to be directed to non-statutory subject matter. More specifically, the Office Action alleges that the claims 1-7 "consist solely of computer program, which is non-statutory functional descriptive material." The Applicant respectfully traverses this rejection for the following reasons.

Claims 1-7 are directed to, in part, a *system* comprising a *pluggable security policy enforcement module.* These claims should therefore be classified as statutory product claims. For instance, § 2106 of the MPEP (page 2100-13 of the May 2004 revision) states, in discussing functional descriptive material, that, "When a computer program is recited in conjunction with a physical structure, such as a computer memory, Office personnel should treat the claim as a product claim." It is true that *one* exemplary and non-limiting way of implementing the pluggable security policy enforcement module is using software; however, in accordance with the MPEP, software is not being claimed *per se* in a proscribed descriptive manner. For the above-identified reasons, the Applicant respectfully requests that the rejection of claims 1-7 be withdrawn.

Regarding the 35 U.S.C. § 102 Rejection

Claims 1-5, 8-14, 16, 19-23, 26-29, 31, 32 and 34-39 were rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,047,377 to Gong (referred to below as "Gong"). Applicant respectfully traverses this rejection for the following reasons.

Gong's invention is primarily directed to security concerns that arise when a computer system is caused to execute software that potentially performs harmful actions on the computer system (col. 1, lines 34-26). More specifically, Gong's invention is directed to problems that arise in systems that allow execution of software from remote sources (col. 6, lines 11-19), including both trusted and untrusted remote sources.

Gong addresses these problems, in part, by implementing security policies that allow trusted code to access more system resources than untrusted code (col. 15, lines 38-41). More particularly, Gong's solution makes use of a permission super class, from which subclasses may be created. Objects that belong to subclasses of the permission super class represent permissions, referred to as permission objects. The permission subclasses inherit the methods and attributes of the permission super class, including a validation method. As the patent states, as the security needs of the system change, the system allows easy modification to adapt to the changes, without requiring specialized knowledge of complex security-management techniques. See, generally, col. 6, lines 11-54 of Gong. Fig. 4 and the accompany discussion of Gong (starting at col. 13, line 57) provide more details regarding the implementation of the above-described features.

In its amended form, claim 1 of above-captioned application combines the subject matter of previously recited claims 1 and 3. As amended, claim 1 recites a system comprising a pluggable security policy enforcement module configured to be replaceable in the system and to provide different granularities of control for a business logic in the system, wherein the business logic processes requests submitted to the system. This claim further recites that the pluggable security policy enforcement module is configured to determine, for a particular granularity of control, whether to permit an operation, requested by a user based at least in part on a permission assigned to the user.

Gong does not disclose each and every feature of claim 1, and, indeed, is directed to a markedly different system than that recited in claim 1. For instance, Gong does not describe at least "a pluggable security policy enforcement module configured to be replaceable in the system and to provide different granularities of control for *a business logic in the system, wherein the business logic processes requests submitted to the system*," and where "the pluggable security policy enforcement module is further configured to determine, for a particular granularity of control, *whether to permit an operation, requested by a user based at least in part on a permission assigned to the user*." Namely, as explained above, Gong provides security provisions to prevent code from trusted and untrusted sources from accessing resources that might cause damage to a computer system. The security provisions in Gong therefore act as a gatekeeper to prevent such *code* from accessing forbidden resources. In contrast, claim 1 sets forth a system in which *business logic* receives requests submitted to the system by a *user*, and it is the role of the pluggable security policy enforcement module to govern a user's interaction with the business logic based at least in part on a permission assigned to the user. Gong does not remotely pertain to the kind of layered environment recited in claim 1, involving the user, business logic, and pluggable security policy enforcement module.

Indeed, Gong and the present invention address entirely different objectives based on underlying different problems. Namely, Gong is primarily concerned with the impact of operations performed by the code itself, presumably independent of the business context in which the code is being used. In marked contrast, the pluggable security module of the present invention is intended to govern a user's interaction with business logic, rather than monitoring the integrity of the business logic *per se* (which can be assumed to originate from a trusted source that does not require checking). More simply stated, Gong is interested in policing code; the invention of claim 1 is interested in regulating a user's interaction with business logic.

In the Office Action, the Examiner cites col. 2, lines 23-31 of Gong, which describes Gong's problem space as "establishing a complex set of relationships between principals and permissions." However, Gong goes on to define "principals" as "processes, objects and threads," rather than the identity of users *per se*. This is consistent with Gong's above-described overarching objective of preventing code from untrusted remote sources from inappropriately accessing system resources, rather than governing the activities of individual users qua users in interacting with business logic.

For at least the above two reasons, the Applicant submits that Gong does not anticipate claim 1. Namely, as stated in MPEP § 2131, a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference. *Verdegaal Bros. v. Union Oil Co. of California*, 2 USPQ2d 1051 (Fed. Cir. 1987). Since Gong does not set forth each and every feature, it fails to anticipate claim 1 under § 102. Moreover, for the reasons stated above, Gong discloses a very different system than the system recited in claim 1, and therefore also does not render claim 1 obvious under 35 U.S.C. § 103.

Claims 2 depends from claim 1, and is therefore allowable for at least this reason.

Claim 4 previously depended from claim 1, but has been rewritten herein into independent form. This claim recites a system comprising a pluggable security policy enforcement module configured to be replaceable in the system and to provide different granularities of control for a business logic in the system, wherein the business logic processes requests submitted to the system. This claim further recites that the pluggable security policy enforcement module includes a control module configured to determine whether to permit an operation based at least in part on accessing the business logic to identify one or more additional tests to perform, and further configured to perform the one or more additional tests.

Gong does not disclose at least a pluggable security policy enforcement module that "includes a control module configured to determine *whether to permit an operation based at least in part on accessing the business logic to identify one or more additional tests to perform*, and further configured to perform the one or more additional tests." In rejecting this feature (which was previously presented in claim 4), the Office Action draws the Applicant's attention to col. 12, line 50 *et seq.* of Gong. That section of Gong describes the use of a non-final method referred to as AdditionalCheck. The AdditionalCheck method may be overridden by a library user to perform an additional security check. As the patent states, because the library user is allowed to override the AdditionalCheck method, the library user has the flexibility to implement security rules that are more restrictive than the original class logic rules (col. 13, lines 52-55). However, while the AdditionalCheck method invokes an additional check, it does not do so by *accessing the business logic to identify one or more additional tests to perform*. Once again, the objective in Gong is to ensure that code that is received from a source does not access system resources in an inappropriate manner. It would therefore be contrary to Gong's design objective to defer to the code itself to determine what additional tests should be performed. In other words, since the objective of Gong is to independently assess the trustworthiness of code, it would seemingly not be appropriate to delegate this question to the code itself (which potentially could have malicious content).

For at least the above two reasons, the Applicant submits that Gong does not anticipate or render obvious claim 4. Claim 5 depends from claim 5, and is therefore allowable for at least this reason.

The other independent claims rejected under § 102 recite, in various permutations, one or more features that are related to the features described above, and are therefore allowable for reasons similar to those given above. For instance, independent claim 8 recites, in part, "checking whether to access a business logic in order to generate a result for the requested operation." At least this feature is not disclosed in or suggested by Gong. Namely, as mentioned, Gong's AdditionalCheck method does not check whether to access business logic in the above-described manner. Even more fundamentally, Gong does not and can not disclose the recited interaction between requested operations, business logic and the pluggable rules set forth in claim 8 because Gong does not employ this kind of layered design paradigm.

Independent claim 19 recites, in part, "determining, based at least in part on a permission assigned to a user, whether to permit an operation based on a request by the user." At least this feature is not disclosed in or suggested by Gong. As mentioned, Gong does not describe a security paradigm based on assigning permissions to users.

Independent claim 26 recites, in part, "checking whether a user requesting to perform the operation is entitled to perform the operation based at least in part on the set of low-level rules." At least this feature is not disclosed in or suggested by Gong. That is, Gong does not describe a security paradigm based on checking whether a *user* can perform an operation; rather, Gong is concerned with whether *code* is permitted to access resources.

Independent claim 31 recites, in part, "assigning high level security concepts to an application domain," and "allowing a set of pluggable rules to define low-level rules, in terms of the high level security concepts, for different business logic in the application domain." At least this feature is not disclosed in or suggested by Gong. As described above, Gong discloses a permission super class, from which various subclasses may depend. But Gong's super class or subclasses merely implement the use of object oriented techniques to structure security functionality, where the purpose of this security functionality is to control the manner in which code accesses system resources. Gong's classes and subclasses are not directed to "different business logic" in an "application domain," as recited in claim 31, but rather different levels of abstraction of security functionality.

Finally, independent claim 35 recites, in part, that "*a business logic layer* to process, based at least in part on the plurality of resources, requests received from a client," and "a pluggable security policy enforcement module to enforce security restrictions on accessing information stored at the plurality of resources." At least this feature is not disclosed in or suggested by Gong for the reasons stated above. Namely, Gong uses his security provisions to verify the integrity of code *per se*, not to interact with a business logic layer that receives requests from a client within a particular business context. More fundamentally, Gong does not employ the kind of multilayered approach set forth in claim 35, involving a tiered interactive relationship involving a client, business logic layer and the pluggable security policy enforcement module.

The various remaining dependent claims rejected under § 102 are allowable at least by virtue of their dependency on the above-identified independent claims.

For at least the above-stated reasons, the Applicant respectfully requests the withdrawal of the 35 U.S.C. § 102 rejection.

Regarding the 35 U.S.C. § 103 Rejection

Claims 6, 7, 15, 17, 18, 24, 25, 30 and 33 were rejected under 35 U.S.C. § 103 as being unpatentable over Gong in view of U.S. Patent No. 5,265,221 to Miller (referred to below as "Miller"). Applicant respectfully traverses this rejection for the reasons stated below.

Miller describes an access control mechanism apparatus 200 as shown in Fig. 2, including a subject memory 204, definition memory 212, rule memory 210, verb memory 208, and object memory 206. The access control mechanism apparatus 200, via an evaluator 202, mediates access of entities by users, providing a YES or NO answer to the question: "Can this USER VERB this OBJECT." The user requests access through input 222. See, generally, col. 3, line 58 to col. 4, line 54 of Miller.

In contrast, representative claim 6 (which has been recast into independent form herein) recites a system comprising a pluggable security policy enforcement module configured to be replaceable in the system and to provide different granularities of control for a business logic in the system, wherein the business logic processes requests submitted to the system. The claim further recites that the different granularities of control comprise a plurality of sets of rules, and wherein each set of rules includes a plurality of permission assignment objects, wherein each of the permission assignment objects associates a user with a particular role, wherein each particular role is associated with one or more permissions, and wherein each of the one or more permissions identifies a particular operation and context on which the operation is to be performed.

First, Gong and Miller do not disclose a pluggable security policy module that controls business logic (where the business logic processes requests submitted to the system). Namely, as stated above, Gong controls the execution of code *per se*, but does not control the operation of business logic that processes requests submitted to the system. Miller's access control mechanism apparatus 200 provides user-based access, but this apparatus 200 does not interact with business logic in the manner recited above. For instance, Miller shows the access control mechanism apparatus 200 interacting with user input module 222; this arrangement in no way suggests the layered approach of claim 6, where a pluggable security policy enforcement module controls business logic, where the business logic processes requests submitted to a system.

Second, the combination of Gong and Miller does not disclose different granularities of control which comprise "a plurality of sets of rules, and wherein each set of rules includes a plurality of permission assignment objects, wherein each of the permission assignment objects associates a user with a particular role, wherein each particular role is associated with one or more permissions, and wherein each of the one or more permissions identifies a particular operation and context on which the operation is to be performed." Namely, as stated above, Gong does not mediate access to resources based on user-related criteria, and therefore does not describe the structure recited in claim 6. Miller does grant access based, in part, on a subject memory 204, but the combination of Gong and Miller does not otherwise remotely suggest the subject matter described in claim 6. Namely, claim 6 does not simply list a laundry list of criteria that play a part in a security check, but recites a specific data structure having a prescribed *organization* of interrelated fields. For frame of reference, Fig. 7 of the instant application describes a data structure which is encompassed by the language used in claim 6 (but does not otherwise limit the scope of the subject matter of claim 6); as shown there, the fields are related together to form a specific data structure. There is no disclosure in the combination of Gong and Miller that Miller's subject memory 204, object memory 206, verb memory 208 and rule memory 210 contain fields that are interrelated in the specific manner recited in claim 6.

Since the combination Gong and Miller fails to disclose each of the features recited in claim 6, it fails to render the subject matter of claim 6 obvious under § 103. As stated in MPEP § 2143.01, to establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 180 USPQ 580 (CCPA 1974). Moreover, there is no suggestion to combine Gong and Miller. As noted above, Gong is foremost concerned with preventing code from accessing unauthorized resources, and is apparently indifferent as to *who* is accessing the resources (perhaps assuming that the user is implicitly trusted). Miller is foremost concerned with protecting users from accessing unauthorized resources, and is apparently indifferent to the *code* that is being executed in connection with such resource access (perhaps assuming that the business code is implicitly trusted). Because of these different design philosophies, one having ordinary skill in the art would not look to Miller to supplement Gong, nor to Gong to supplement Miller.

The other claims rejected claims under § 103 recite related subject matter to claim 6 (or are dependent from such related claims), and are therefore allowable over the combination of Gong and Miller for reasons similar to those given above.

For at least the above-stated reasons, the Applicant respectfully requests the withdrawal of the 35 U.S.C. § 103 rejection.

Regarding the Newly Added Claims

The newly added claims, 40-42, depend from independent claim 1. These claims are therefore allowable for at least the reasons given in connection with claim 1. Moreover, the new claims recite subject matter which further distinguishes the claimed invention over the combination of Gong and Miller.

For instance, claim 40 recites that the system is configured as a multi-layer architecture, wherein the business logic is implemented as a business logic layer of the multi-layer architecture. Neither Gong nor Miller, whether considered alone or in combination, disclose or suggest this additional subject matter.

Claim 41 recites that the pluggable security policy enforcement module is configured to receive an input from the business logic in the form of a user indication and an item indication. Neither Gong nor Miller, whether considered alone or in combination, disclose or suggest this additional subject matter.

Claim 42 recites that the pluggable security policy module includes an interface that provides the following interface functionality: first functionality for testing whether an identified item can be approved by a specified user; second functionality for testing whether the identified item of a specified type can be created by the specified user; third functionality for testing whether the identified item can be deleted by the specified user; fourth functionality for testing whether the identified item can be modified by the specified user; and fifth functionality for testing whether the identified user can examine details of the identified item. Neither Gong nor Miller, whether considered alone or in combination, disclose or suggest this additional subject matter.

Cross Reference to Commonly Assigned Applications

The following commonly-assigned applications were filed on the same date as the present application: 09/847,063; 09/845,752; 09/845,751; 09/847,067; 09/845,737; 09/845,780; 09/847,038; and 09/847,035

Conclusion

The arguments presented above are not exhaustive; Applicant reserves the right to present additional arguments to fortify its position. Further, Applicant reserves the right to challenge the alleged prior art status of one or more documents cited in the Office Action.

All objections and rejections raised in the Office Action having been addressed, it is respectfully submitted that the present application is in condition for allowance and such allowance is respectfully solicited. The Examiner is urged to contact the undersigned if any issues remain unresolved by this Amendment.

Respectfully Submitted,

Dated: 2/3/2005 By: ______________________

David M. Huntley
Reg. No. 40,309
(509) 324-9256